

REMARKS

Reconsideration and allowance of the present application are respectfully requested. Claims 1-58 are currently pending in this application.

I. Preliminary Comments Regarding the Approach Taken by this Response

This Response addresses the outstanding Office Action of November 30, 2005. The November 30, 2005 Office Action incorporates by reference parts of the first Office Action of March 10, 2005 without expressly repeating the complete content of those rejections (e.g., principally with respect to the 35 U.S.C. § 112 rejections and the objections to the specification and drawings). Therefore, where applicable, this Response also addresses parts of the March 10, 2005 Office Action. And for this reason, certain arguments presented in this paper take the form of a repetition of the arguments presented in the September 12, 2005 Response, as supplemented by new arguments which address the Examiner's comments in the November 30, 2005 Office Action (e.g., principally with respect to the "Response to Arguments" section of the outstanding Office Action which begins on page 2 of the Office Action).

This Response is accompanied by two appendices. A first appendix, Appendix A, provides a chart which lists the pending claims and which correlates the features in the claims with portions of the specification and drawings which support the features. It should be noted that any reference to the specification and drawings in this Response is made to assist the Patent Office in the examination of this application and to provide a comprehensive response to the various rejections and objections identified in the outstanding Office Action. However, these references *should not be construed as limiting the scope* of the claims. Moreover, the chart's references to the specification and drawings reflect only representative support for the features in the specification and

drawings; in other words, the chart does not exhaustively list all of the parts of the specification and drawings which support the features. A second appendix, Appendix B, provides an exhibit which pertains to the meaning of the term "layer" in the computing arts.

This Response is also accompanied by a Supplemental Information Disclosure Statement.

II. Regarding the 35 U.S.C. § 112, First Paragraph, Rejection

Claims 1-30 and 48-51 were rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the enablement requirement. The Applicant respectfully traverses this rejection for the following reasons.

As a preliminary matter, as set forth in MPEP § 2164.01, the overarching question in any § 112, first paragraph, analysis pertaining to enablement is whether the disclosure, when filed, contains sufficient information to enable one skilled in the pertinent art to make and use the claimed invention without undue experimentation. The MPEP also identifies, in § 2164.04, the due process procedure to be applied by the Office to establish a *prima facie* case of lack of enablement:

In order to make a rejection, the examiner has the initial burden to establish a reasonable basis to question the enablement provided for the claimed invention. *In re Wright*, 999 F.2d 1557, 1562, 27 USPQ2d 1510, 1513 (Fed. Cir. 1993) (examiner must provide a reasonable explanation as to why the scope of protection provided by a claim is not adequately enabled by the disclosure). A specification disclosure which contains a teaching of the manner and process of making and using an invention in terms which correspond in scope to those used in describing and defining the subject matter sought to be patented must be taken as being in compliance with the enablement requirement of 35 U.S.C. 112, first paragraph, unless there is a reason to doubt the objective truth of the statements contained therein which must be relied on for enabling support. Assuming that sufficient reason for such doubt exists, a rejection for failure to teach how to make and/or use will be proper on that basis. *In re Marzocchi*, 439 F.2d 220, 224, 169 USPQ 367, 370 (CCPA 1971). As stated by the court, "it is incumbent upon the Patent Office, whenever a rejection on this basis is made, to explain why it doubts the truth or accuracy of any statement in a supporting disclosure and to back up assertions of its own with acceptable evidence or reasoning which is inconsistent with the contested statement. Otherwise, there would be no need for the applicant to go to the trouble

and expense of supporting his presumptively accurate disclosure.” 439 F.2d at 224, 169 USPQ at 370.

To summarize this passage, a specification is presumptively considered to comply with the enablement requirement. It is the obligation of Patent Office to “to **explain** why it doubts the truth or accuracy of any statement in a supporting disclosure and to **back up assertions of its own with acceptable evidence or reasoning which is inconsistent with the contested statement** (emphasis added).”

In the present case, the Applicant has submitted a specification of 102 pages of text, along with 23 figures. As identified in detail in the chart presented in Appendix A, the claims in the application recite elements that are explained in the specification and drawings, and the claims adopt terminology that is consistent with the terms used in the specification and drawings. In accordance with the MPEP, this lengthy specification presumptively complies with the enablement requirement of 35 U.S.C. § 112, first paragraph. It is the general position of the Applicant, as explained in detail below, that the Office Action communicates the Patent Office’s *conclusion* that the application does not satisfy the enablement requirement, but the Office Action fails to also provide, with any meaningful degree of specificity, an *explanation* of why the application is deemed deficient. To be more precise, to broadly say that an application fails to adequately disclose elements X, Y, and Z is not sufficient where the application does, in fact, devote considerable explanation to such features; in accordance with the MPEP passage cited above, the Office bears the initial burden of addressing the content of the application’s disclosure of these features and *explaining* why the disclosure is deficient with “acceptable evidence or reasoning.” In other words, the MPEP requires a reasoned engagement with what the specification sets forth. Since the Office Action does not

provide such explanation, the 35 U.S.C. § 112, first paragraph, rejection fails to shift the burden of rebuttal to the Applicant.

Now addressing the specifics of the § 112, first paragraph, rejection, the paragraph No. 4, paragraph No. 5 of the March 10, 2005 Office Action states:

The Examiner submits that the specification and claims go into great detail on the technical aspects of the invention. But the Examiner strongly believes that the invention cannot be enabled because the Examiner cannot ascertain what the invention is attempting to accomplish. The specification refers to various modules, which are swapped out to adapt the architecture to different domains. [page 6] The Examiner requests that Applicant clearly cite where in the specification these various modules are defined and what they actually accomplish. The Examiner sees that Applicant refers to the use of this architecture in various domains, and gives examples of such. The Examiner is unclear how, if at all, the invention can be implemented to actually work in said various domains. The Examiner understands that Applicant wishes to receive all possible breadth of claim coverage. In this case, the Applicant has attempted to describe vaguely a multiple of uses for the invention with a very clear technical description of the underlying subject matter. Unfortunately this combination has made it very difficult to actually implement the invention because there is no hint of how to implement the technical disclosure. A broader, over simplified analogy could be made to a person who has being [sic] given a detailed schematic of a telephone wiring closet without saying why it is there, what the telephone wiring closet actually is, or how it can be used to connect telephones so people can talk to each other over them. That person would then be able to read and understand the schematic, but would suffer an unreasonable burden in attempting to grasp what the intention is of the device described in the schematic and then once the intention was discovered would suffer yet another unreasonable burden in actually deciding how the invention could be applied to various environments and implementing the invention to fit that inferred application.

Thus, part of the rejection is based on the Examiner's perception that the application fails to teach how to use the invention – e.g., because the “Examiner cannot ascertain what the invention is attempting to accomplish.” Another part of the rejection is based on the Examiner's assertion that the application fails to disclose how to make the

invention – e.g., because the “Examiner is unclear how, if at all, the invention can be implemented to actually work in said various domains.” The Applicant respectfully disagrees with the Patent Office’s position for the reasons which follow.

As to the question of “what the invention is attempting accomplish,” the present specification is abundantly clear. While the invention does not address a “univocal” need, and therefore does not provide a “univocal” solution, *one* prominent goal of the subject matter described in the specification is to provide a server architecture that can accommodate modification in an efficient and flexible manner.

Consider the problem addressed on page 1 of the specification. As indicated there, “source code is typically developed specifically for the domain of one computer application.” Since “the source code is developed specifically for each domain, the components of one computer application developed specifically for one domain might not be reusable for another computer application under development for another domain.” As a further result, “because of the inability to reuse high-level components for multiple computer applications across diverse domains, the cost of developing a computer application can be quite high. In addition, because the components are new, their reliability is often unproven.”

As set forth in the Summary section on pages 3 and 4, the subject matter disclosed in the present application addresses at least the above-identified problem through the use of a multi-layer server architecture. One exemplary benefit of the multi-layer architecture is set forth in page 4, lines 6-10:

Any one of the layers may be removed, modified, or updated without impacting other layers. This allows the architecture to adapt easily to many different problem domains, to support many different types of client devices, to accommodate many different users in different regions and cultures of the world, and to interface with many diverse resources.

Further discussion regarding the flexibility of the architecture can be found at least on page 6, lines 5-21, and on page 9, lines 11-21 of the specification.

Thus, in answer to the Office Action's question of "what does the invention accomplish," the Applicant submits that one exemplary and non-limiting objective of the subject matter described in the present application is to provide a modular architecture that readily allows logic to be swapped in and out, thus expediting software development. The specification is abundantly clear on this issue.

In reply to this argument, paragraph No. 2 of the outstanding Office Action states in part:

The specification provided by Applicant is not 'abundantly clear' about the intended goal of the invention. Applicant has stated that the invention's purpose is 'to provide a server architecture that can accommodate modification in an efficient and flexible manner.' Applicant is invited to explain this terminology in a manner which one of ordinary skill in the art would be able to grasp the goal of the invention. As it stands, Applicant has stated an unclear purpose for the invention and one of ordinary skill in the art would not be able to ascertain what the invention accomplishes, which is necessary for implementation of the invention given the state of the specification.

Here, the Office Action is repeating an introductory sentence in the previous Response of September 12, 2005, repeated above in the present Response. However, the Applicant points out that the previous Response did not limit its explanation to this single sentence, but continued its explanation, starting with the next paragraph, i.e., the paragraph which begins "Consider the problem . . ." This explanation is also repeated above in this Response. The Patent Office fails to explain *what* is not clear about the *complete* explanation provided above. More generally, at least the first ten pages of the specification adequately set forth some of the goals of the present invention. The Patent

Office is requested to engage this text and *explain why* it is deficient; simply asserting that it is deficient fails to establish a *prima facie* case of lack of enablement.

The outstanding Office Action also addresses the above-identified argument by stating that the specification does not explain what is meant by a problem domain (e.g., note paragraph No. 3 of the outstanding Office Action). However, even page 1 of the specification defines a domain as follows:

Domains pertain to a particular category or area of service that the application provides. Example domains include asset management, leasing and lending, insurance, financial management, inventory tracking, resale and repair management, and so forth

The Patent Office has not established why this explanation would not be understood by one skilled in the art.

Returning to passage quoted above from March 10, 2005 Office Action, the telephone wiring closet analogy addresses the “use” component of the enablement question in a different manner by implying that the specification fails to inform a practitioner in the relevant art how to actually use the multi-layer architecture in a concrete context. Again, this position is misplaced because the specification is abundantly clear regarding the manner in which the invention can be used. Consider, for example, the exemplary and non-limiting application shown in Fig. 1. Fig. 1 shows a network system 100 in which the tiered software architecture may be implemented. The system 100 includes multiple clients 102(1), 102(2), 102(3), ..., 102(N) that submit requests via one or more networks 104 to an application server system 106. Upon receiving the requests, the server system 106 processes the requests and returns replies to the clients 102 over the network(s) 104. In some situations, the server system 106 may access one or more resources 108(1), 108(2), ..., 108(M) to assist in preparing the replies.

Simply put, in one exemplary and non-limiting implementation, the multi-layer architecture can be employed to field user queries in an online manner. This pertains to a "real world" application, steering one skilled in the art to one exemplary practical application of the concepts described in the presentation application.

In paragraph No. 6 of the Office Action, the Patent Office states:

Applicant argues again that one of ordinary skill in the art would be able to implement the invention. It is clear that one of ordinary skill in the art would not be able to implement the invention. The allegation by Applicant that the invention can be utilized in an online manner to field user queries is not adequately supported or explained by the specification in sufficient detail to allow one of ordinary skill in the art to actually implement the invention.

To repeat, Fig. 1 and the accompanying discussion in the specification establish the application of the architecture to an online environment, and the remaining figures and associated discussion explain in great detail how the architecture can be constructed according to one exemplary implementation. This disclosure is presumptively enabling. The Patent Office has not met its initial burden by engaging the specification and explaining why the disclosure is deficient. Paragraph No. 6 of the outstanding Office Action conveys a conclusion, not an explanation.

Now advancing to the question of *how* to make the invention, the legal scope of the invention, of course, is defined by the claims. Consider claim 1, which recites, *inter alia*, the elements of a multi-layer application that includes a problem-solving logic layer, an execution environment layer, an interface layer, and a presentation layer. The specification sets forth (with reference to Fig. 2) an architecture having a hierarchy of modules, including a business logic layer 204, an execution environment 202, various interface layers (e.g., 206, 208, etc.), a presentation layer 212, and so forth. The architecture shown in Fig. 2 is described at least on page 8, line 20 to page 15, line 26 of

the specification. Fig. 3 illustrates one exemplary manner of operation of architecture shown in Fig. 2. The procedure shown in Fig. 3 is described at least on page 16, line 1 to page 19, line 2 of the specification. Also note Appendix A which establishes, in greater detail, exemplary support for what is being claimed in the specification and drawings.

In further regard to the question of how to make the invention, the Office Action questions how “the invention can be implemented to actually work in said various domains.” The specification thoroughly describes one exemplary and non-limiting application of the architecture to the field of asset management. Note, for instance, page 19, line 5 to page 33, line 1 of the specification. (As to this matter, also note that the claims are not presently directed to any particular business domain *per se*, thus reducing the need to exhaustively describe many different business domains. In other words, the subject matter described in the specification is directed to an architecture which can accommodate different business models, rather than the composition of business-specific models *per se*.)

The outstanding Office Action addresses selected points in the above-identified arguments. For instance, paragraph Nos. 4 and 5 of the outstanding Office Action assert, in essence, that the specification fails to provide sufficient detail to enable one skilled in the art to produce the kind of code flexibility mentioned above, in which code modules can be swapped in and out without impacting other code modules. Again, the Applicant has presented 102 pages of text and 23 drawings that explain how to achieve this kind of flexibility. The chart in Appendix A further assists the Patent Office by identifying exemplary support for what is being claimed in the specification and drawings. The Patent Office has not explained with any meaningful specificity why the disclosure provided in the specification is deficient. To say that “Applicant has submitted a large amount of technical and architectural information and neglected to explain the

implementation of the technical and architectural information to one or ordinary skill in the art” (in paragraph No. 5) conveys a broad conclusion, not an explanation of why the 102 pages of text in the specification are deficient. Moreover, the Applicant submits that the so-called “technical and architectural information” alluded to in the outstanding Office Action specifically address how to implement the invention (according to one exemplary and non-limiting embodiment), and is not tangential to this topic. Indeed, it is not clear what alternative role the Patent Office believes this detail serves.

To address the Office Action’s statements more directly, two exemplary features of the architecture which contribute to the goal of “flexibility” are summarized as follows. First, the presentation layer 212 is configured to interact with different client devices 102 without requiring the business logic 204 to specifically account for these devices 102. Second, the interface layer (e.g., the data coordination layer 206 and the data abstraction layer 208) allows the business logic 204 to interact with different kinds of resources 108 without requiring the business logic 204 to include specifically tailored code to deal with these resources. (These two features are merely representative; yet other disclosed factors contribute to the flexibility of the multi-layer application.) Further note the chart provided in Appendix A for a more detailed correlation of the features as claimed with representative portions of the specification and drawings which support the features.

Next, in paragraph No. 7 of the outstanding Office Action, the Patent Office states that the “interface layer,” which is included as an element in claim 1, is not disclosed in the specification. The Applicant submits that this feature is disclosed in the specification. For reference, claim 1 specifically recites, in part:

an interfacing layer to interface the problem-solving logic layer with one or more resources so that the execution models may utilize the resources when processing the client requests

The specification and drawings show various layers which interface the problem-solving logic layer with the resources, such as the data coordination layer 206 and the data abstraction layer 208. Moreover, original claim 10 expressly identifies the interfacing layer with the data coordination layer 206 and the data abstraction layer 208, *leaving no doubt* as to the kinds of layers that the term “interfacing layer” is intended to encompass. Finally, to more fully address the Office Action’s comment, the specification has been amended to include the same language used in claims 1 and 10 in the Detailed Description section. For any one or more of the above reasons, the Applicant submits that the specification adequately explains what is meant by the interfacing layer and how to implement the interface layer (as identified, for example, in the chart in Appendix A).

Finally, in paragraph No. 8 of the outstanding Office Action, the Patent Office states that, contrary to the argument above, pages 19-33 do not describe how one skilled in the art can implement the invention in the field of asset management. To repeat, for the Patent Office to state that over ten pages of technical disclosure (pertaining to this feature) is not enabling is a conclusion; such a statement does not engage with what is disclosed to provide an explanation of why it is allegedly deficient. Further, the Office Action questions what is meant by asset management. As stated on page 20, lines 13-15 of the specification, an asset catalog application “allows a user to view, create, and modify information related to assets (e.g., products) stored in an electronic catalog.” This is what is meant in this Response by asset management.

As a final point, the Applicant points out that this application is a member of a family of nine patent applications having the same filing date (April 30, 2001). These applications include Application Nos. 09/845,751, 09/845,752, 09/845,780, 09/847,035,

09/847,037, 09/847,038, 09/847,063, and 9/847,067. These applications include similar terminology to the present application, and use the same introductory figures, such as the multi-layer architecture drawing shown in Fig. 2 of the presentation application. As per the typical course, some of the Examiners handling this family of applications raised minor 35 U.S.C. 112, second paragraph, rejections. Further, the Applicant submitted one amendment in this family of applications which provoked a 35 U.S.C. § 112, first paragraph, rejection. However, none of the Examiners have raised the kinds of global 35 U.S.C. § 112, first paragraph, concerns identified in the present Office Action. This is highly significant, as it indicates that others were not similarly stymied by the kind of exposition provided by the present application. This observation at least has a bearing on how the present application would be interpreted by one skilled in the pertinent art.

The outstanding Office Action addresses to point as follows (in paragraph No. 11 of the Office Action):

The Examiner of record is unaware of any citation within the MPEP that states the Examiner is bound by the work of another Examiner on a co-pending application. Further, the co-pending applications were not combined *en masse* and presented to any other Examiner in an omnibus format. This observation also has a bearing on how the present application would be interpreted by one of ordinary skill in the pertinent art.

First, there is no attempt to say that another Examiner's position binds the present Examiner; it does not. Rather, the argument is that the positions taken by other Examiners constitute *evidence* that the specification is enabling. Second, the fact that the present specification includes several features does not introduce confusion because, for instance, the specification takes care to organize the subject matter into discrete, yet interrelated, sections. Moreover, many of the Office Action's criticisms in this application concern terminology, and the question of the comprehensibility of such

terminology is largely independent of whether an application discloses ten features or a thousand features.

For the above reasons, the specification satisfies 35 U.S.C. § 112, first paragraph, by explaining how to make and use the invention, without requiring the exercise of undue experimentation. Accordingly, the Applicant respectfully requests the Patent Office to withdraw the 35 U.S.C. § 112, first paragraph, rejection.

III. Regarding the 35 U.S.C. § 112, Second Paragraph, Rejection

Claims 1-30 and 48-51 were rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The Applicant respectfully traverses this rejection for the following reasons.

Paragraph No. 10 of the March 10 Office Action broadly states that many terms used in the claims are indefinite. More specifically, the Office Action states that:

With regard to claims 1-53, Applicant has freely acted as his own lexicographer. Applicant has used multiple terms that are not well known in the art and the Examiner has not encountered satisfactory definitions for said terms within the specification or within the prior art, including multiple technical and computer dictionaries, as to clear up this deficiency. The terms are indefinite because the specification does not clearly define the terms to the extent required by the Examiner in order to allow one of ordinary skill in the art to reasonably understand the specification and invention with ease and clarity.

The Office Action also identifies no fewer than 42 terms that are believed to be indefinite. And the Office Action qualifies the list of 42 terms by stating that the list is only a representative list.

The Office Action's legal reasoning is misplaced. Unique inventions are, by definition, new, thus sometimes warranting the use of terms that are not found in technical dictionaries. With respect to these terms, MPEP § 2173.05(a) states:

New terms are often used when a new technology is in its infancy or is rapidly evolving. The requirements for clarity and precision must be balanced with the limitations of the language and the science. If the claims, read in light of the specification, reasonably apprise those skilled in the art both of the utilization and scope of the invention, and if the language is as precise as the subject matter permits, the statute (35 U.S.C. 112, second paragraph) demands no more. *Shatterproof Glass Corp. v. Libbey Owens Ford Co.*, 758 F.2d 613, 225 USPQ 634 (Fed. Cir. 1985)

More specifically, in the logic-implemented arts, an element of an invention often constitutes a module within a software program, corresponding to a collection of code instructions. Because code performs a function, a specification can often fully describe such a module by setting forth the function that the module performs. The name assigned to such a module is often merely a label of convenience. In other words, because the module has no counterpart in terms of what came before, the person drafting the application will assign an arbitrary (but preferably descriptive) name to the module as simply a convenient reference. Consider the hypothetical case where the specification satisfactorily describes an XYZ module which performs function ABC, where "XYZ" is an arbitrary label assigned to this module. In this case, in accordance with accepted claim drafting standards, a claim that recites "an XYZ module configured to perform ABC" would pass muster under 35 U.S.C. 112, second paragraph. The Office Action reads as if the specification is rendered vague because the terms used in the claims cannot be found in a technical dictionary. But as stated above, this position applies a standard that is legally inappropriate.

Applying these general guidelines to the present application, Applicant has taken care to use terms in the specification and the claims which are, in themselves, inherently descriptive. Thus, many of the terms used in the claims should be clear based on the plain meaning of the terms alone. For example, the Examiner questions what the term "multi-layer application" means. A multi-layer application is an application having multiple layers. Insofar as the textual portion of the specification consistently uses this term, and the drawing portion of the specification illustrates the concept of a multi-layer application (e.g., with reference to Fig. 2), then the specification satisfies 35 U.S.C. § 112, second paragraph.

Moreover, many other terms identified in the Office Action are used in the claims in such a manner that, upon introduction of a term, the *claim itself*, when read in light of the specification, defines what is meant by the term. To take one example, the Office Action questions what is meant by "interfacing layer" in claim 1. Claim 1 recites "an interfacing layer to interface the problem-solving logic layer with one or more resources." Implicit in this recitation is the definition that the interfacing layer is a layer which interfaces the problem-solving logic layer with one or more resources. This is abundantly clear on its face and is consistent with the usage of this term in the specification. The law demands no more. The other terms identified in the Office Action can be considered definite for similar reasons.

In paragraph No. 14 of the outstanding Office Action, the Patent Office addresses the above arguments by stating, "The terms within the claims, read in light of the specification, are indefinite to one of ordinary skill in the art." This is a conclusion, not an explanation. For instance, Applicant has pointed to the fact that the terms often incorporate their own respective definitions within the claims, as in the above-described case of the interface layer. The Office Action has not set forth *what* precisely is not

understood about an interfacing layer which interfaces the problem-solving logic layer with one or more resources. This language is unproblematic, identifying a role of the interfacing layer in the context of two other elements recited in the claims (the problem-solving logic layer and the resources). The interfacing layer *is* what it *does*.

In paragraph No. 15, of the outstanding Office Action, the Patent Office states that the term “multi-layer application” would not be understood because the specification does not define the concept of a layer. The term layer is a ubiquitous term in the data processing arts, as exemplified by the exhibit in Appendix B. And even the Stevens reference applied in the Office Action (to be discussed below) uses the term “layer,” and thus is evidence of the well known use of this term in the art. Moreover, the specification, with reference to at least Fig. 2, shows a hierarchal organization of layers, which can be implemented as a hierarchical grouping of code modules. Hence, the term “layer” would be well understood by one skilled in the art.

In paragraph No. 16, the Office Action states that the term “interfacing layer” is not described in the specification. However, the Applicant submits that this term is adequately described in the specification for the same reasons set forth in response to the 35 U.S.C. § 112, first paragraph, rejection.

For the above-identified reasons, the Patent Office is respectfully requested to remove the 35 U.S.C. § 112, second paragraph, rejection.

IV. Regarding the Objections to the Drawings

Paragraph No. 2 of the March 15, 2005 Office Action objects to the drawings. More specifically, this portion of the Office Action reads, in part:

The drawings are objected to because, though the drawings are comprehensive in nature and cover multiple aspects of the invention, the drawings still fail to convey to one of ordinary

skill in the art what exactly is being accomplished by the invention. The closest drawing that the Examiner feels is to showing the actual usage of the invention, which is still unclear, is Figure 20, which shows a login prompt on a web page and a human translator. Even with these two items present in Figure 20, and the descriptions given for this and all other submitted drawings, the Examiner is not assisted in graphing the invention at all based upon the currently submitted drawings.

Again, the legal scope of the invention is defined by the claims. For example, claim 1 recites several layers. These layers find exemplary and non-limiting support in at least Fig. 2 of the present application (note for instance, the above discussion pertaining to the 35 U.S.C. § 112, first paragraph rejection). Further, Fig. 3 describes one exemplary and non-limiting manner of operation of the architecture shown in Fig. 2. Further still, Fig. 1 describes one exemplary and non-limiting application of the invention to an online query-type system. Thus, even this collection of introductory figures satisfactorily addresses the questions raised in the Office Action.

Paragraph No. 20 of the Office Action states that "The objection to the drawings is maintained, since one of ordinary skill in the art cannot use the currently provided drawings to assist in implementation of the invention." However, the Appendix A sets forth how each of the elements recited in the claims finds support in the drawings. The Office Action's position is a conclusion, not an explanation; the Patent Office must explain why 23 figures of drawings are deemed deficient.

Since the drawings clearly illustrate exemplary implementations of the invention, the Applicant submits that no drawing changes are warranted. And for this reason, the Applicant respectfully requests the Patent Office to withdraw the objection to the drawings.

V. Regarding the Request for a Substitute Specification

In paragraph No. 3 on page 3 of the March 10, 2005 Office Action, the Patent Office requests the Applicant to submit a substitute specification pursuant to 37 CFR § 1.125(a). The Office Action states that a substitute specification is required in view of the alleged deficiencies noted in the 35 U.S.C. § 112 rejections. However, as per the arguments presented above, the Applicant submits that the application satisfies both the first and second paragraphs of 35 U.S.C. § 112. Since the specification requires no changes, the Patent Office is respectfully requested to remove the requirement for a substitute specification.

VI. Regarding the 35 U.S.C. § 102 Rejection

Claims 1-9, 12-16, 18-21, 23-24, and 48-51 were rejected under 35 U.S.C. § 102(b) as being anticipated by an excerpt from a book entitled, "UNIX Network Programming," by W. Richard Stevens (referred to below as "Stevens"). Applicant respectfully traverses this rejection for the following reasons. The arguments presented to address the § 102 rejection are to be interpreted as representative rather than exhaustive.

Stevens describes various features of UNIX technology. In pages 334-341, Stevens describes an Internet superserver. The Internet superserver uses a single daemon (inetd) to service multiple connection requests (where a "daemon" refers to a background program). When a request is received pertaining to a particular service, the inetd daemon "forks" to an appropriate child process. The child process handles the service request. In other portions, Stevens discusses a presentation layer (e.g., page 250). The presentation layer modifies the representation of data to facilitate its exchange.

Stevens does not disclose or render obvious the invention recited in the claims.
Consider claim 1, as currently amended, and reproduced in its entirety below.

1. A server system comprising:
 - one or more computers; and
 - a multi-layer application executing on the computers to handle client requests submitted by various client devices, the multi-layer application comprising:
 - a problem-solving logic layer to process the client requests according to an associated problem domain, wherein the problem domain pertains to a particular category of service, the problem-solving logic layer containing one or more execution models to perform various sets of tasks when processing the client requests, the problem-solving logic layer producing replies to the client requests;
 - an execution environment layer to receive the client requests and select an appropriate execution model in the problem-solving logic layer for processing the client requests;
 - an interfacing layer to interface the problem-solving logic layer with one or more resources so that the execution models may utilize the resources when processing the client requests; and
 - a presentation layer to receive the replies produced by the problem-solving logic layer and to structure the replies in a manner that makes the replies presentable on the various client devices,
- wherein any of the layers may be changed without impacting other layers.

For instance, Stevens does not disclose a multi-layer application that handles client requests submitted by various client devices, wherein the multi-layer application includes the claimed combination of a problem-solving logic layer, an execution environment layer, an interfacing layer, and a presentation layer, wherein any of the layers may be changed without impacting other layers.

The outstanding Office Action states (in paragraph No. 36) that Steven's inetd process meets the elements in claim 1 which call for a problem-solving logic layer, an execution environment layer, and an interfacing layer. However, there is nothing in

Stevens that would lead one of ordinary skill in the art to interpret the various aspects of the inetd daemon as forming three distinct layers, where these layers may be changed without impacting other layers. And indeed, it remains unclear the manner in which the Patent Office itself is interpreting the inetd daemon as forming three distinct layers.

Paragraph No. 25 of the outstanding Office Action states that "Applicant's amendment of *any of the layers may be changed without impacting other layers* was supported by Stevens, in that Stevens is a flexible programming structure designed for ease of use by one of ordinary skill in the art." Even if, assuming *arguendo*, it was agreed that Stevens was designed for "ease of use," this does not establish that Steven's inetd process comprises three distinct layers where any of the layers may be changed without impacting other layers.

For at least the above-defined reasons, the Applicant submits that Stevens fails to disclose or render obvious the subject matter recited in independent claim 1.

Now advancing to independent claim 48, this claim is reproduced in its entirety below:

48. A method for processing client requests in a system, comprising:
receiving requests from multiple clients, the requests being related to a business-related problem domain, wherein the business-related problem domain pertains to a particular category of business-related service;
processing the requests within problem-solving logic to produce replies within the business-related problem domain, the processing comprising requesting data to be used in formulating the replies;
retrieving the data from one or more external resources and mapping the data to a domain framework for the business-related problem domain, the domain framework being independent from the problem-solving logic; and
interfacing the problem-solving logic to the domain framework to obtain the data for use in processing the request,
wherein a new business-related problem domain can be exchanged for a previous business-related problem domain by replacing one or more components of the system,

without having to reconstruct an entire application solution for the new business-related problem domain.

Stevens does not disclose the above-described subject matter. For instance, Stevens does not disclose that its UNIX daemons implement business-related problem domains, or that a new business-related problem domain can be exchanged for a previous business-related problem domain by replacing one or more components of the system, without having to reconstruct an entire application solution for the new business-related problem domain.

Further, Stevens does not disclose the operations of “retrieving the data from one or more external resources and mapping the data to a domain framework for the business-related problem domain, the domain framework being independent from the problem-solving logic,” and “interfacing the problem-solving logic to the domain framework to obtain the data for use in processing the request.” Portions such as Section 5.6.6 of Stevens (on page 250) disclose a process for converting from an abstract syntax to a transfer syntax. This functionality, however, does not meet the specific subject matter of claim 48, where data from one or more external resources is mapped to a domain framework for a business-related problem domain, and where the problem-solving logic is interfaced to the domain framework to obtain the data for use in processing the request.

Paragraph No. 48 of the outstanding Office Action states that MPEP § 2111.01 instructs that the broadest reasonable interpretation should be given the claims during examination. However, MPEP § 2131 states that “A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” The Applicant submits that the Office Action has not shown where the Stevens reference meets the elements of claim 48 identified above, and

an interpretation that cannot account for all of the elements in a claim is not a reasonable interpretation.

For at least the above-defined reasons, the Applicant submits that Stevens fails to disclose or render obvious the subject matter recited in independent claim 48.

The remainder of the claims that are rejected under § 102 depend from either independent claim 1 or claim 48. These claims are allowable for at least for this reason. Moreover, these claims recite additional features which are not disclosed in (or rendered obvious by) Stevens.

To cite just one example, dependent claim 20 recites:

20. (Original) A server system as recited in claim 1, further comprising a constraint system to constrain operation of the multi-layer application according to multiple different constraints, the constraint system comprising a hierarchy of constraint layers, with each constraint layer containing a set of one or more constraints that customize operation of the multi-layer application.

Stevens does not disclose or suggest a constraint system that constrains the operation of a multi-layer application, where the constraint system is comprised of a hierarchy of constraint layers. The outstanding Office Action identifies Steven's configuration file (disclosed on pages 335-336) as having a bearing on this feature. However, the configuration file merely specifies the services that the superserver is to listen for and what to do when a service request arrives. It does not comprise a *hierarchy* of constraints. A flat list of information is not a hierarchy. Nor does the information in the configuration file *customize* the operation of an application as claimed. Paragraph No. 27 of the outstanding Office Action states again that the broadest reasonable interpretation is applied to the Stevens reference; but an interpretation that fails to account for the express

elements in the claims (e.g., a *hierarchy* of constraints) cannot be a reasonable interpretation.

For the above reasons, the Patent Office is respectfully requested to withdraw the rejection based on 35 U.S.C. § 102(b).

VII. Regarding the 35 U.S.C. § 103 Rejections

Claim 17 was rejected under 35 U.S.C. § 103 as being unpatentable over Stevens and the Patent Office's "Official Notice." Claims 22 was rejected under 35 U.S.C. § 103 as being unpatentable over Stevens alone. Claims 10-11 were rejected under 35 U.S.C. § 103 as being unpatentable over Stevens in view of an excerpt from a book entitled, "UNIX in a Nutshell," by Daniel Gilly (referred to below as "Gilly"). Claims 25-28 were rejected under 35 U.S.C. § 103 as being unpatentable over Stevens in view of an excerpt from a book entitled, "UNIX Power Tools," by Jerry Peek et al. (referred to below as "Peek"). And claims 29-30 were rejected under 35 U.S.C. § 103 as being unpatentable over Stevens in view of an excerpt from a book entitled, "Creating Worldwide Software," by Bill Tuthill (referred to below as "Tuthill"). Applicants respectfully traverse each of these rejections for the following reasons.

First, the claims rejected under § 103 (claims 10-11, 17, 22, and 25-30) ultimately depend from claim 1. Since the secondary references (Gilly, Peek, and Tuthill) do not remedy the deficiencies noted above with respect to claim 1, then claims 10-11, 17, 22, and 25-30 are allowable for at least this reason.

In addition, claims 10-11, 17, 22, and 25-30 recite additional features which are not disclosed in or rendered obvious by any combination of the applied art – Stevens, Gilly, Peek, and Tuthill. This statement will be demonstrated with respect to

representative claims 10, 17, 22 and 25. As before, the arguments presented to address the § 103 rejection are to be interpreted as representative rather than exhaustive.

Consider first dependent claim 10, which was rejected based on a combination of Stevens and Gilly. It recites:

10. A server system as recited in claim 1, wherein the interfacing layer comprises:
a data abstraction layer to obtain data from the resources and map the data into a domain framework that models information flow for a specific problem domain; and
a data coordination layer that provides an interface for the problem-solving logic layer to access the domain framework of the data abstraction layer and obtain the data.

Neither Stevens nor Gilly disclose an interfacing layer comprising a combination of a data abstraction layer and a data coordination layer as claimed. The outstanding Office Action cites pages 10-1 to 11-11 of Gilly as having a bearing on the subject matter recited in claim 10. This portion of Gilly describes an editor referred to as "sed." While this editor can loosely be said to modify a file, this feature is not even remotely related to an interfacing layer that includes a data abstraction layer and a data coordination layer as claimed.

Paragraph No. 28 of the outstanding Office Action states that the applied art meets the claim elements particularly in view of the specification's failure to disclose an interfacing layer. However, this claim defines an interfacing layer as a combination of a data abstraction layer and a data coordination layer, which the specification and drawings set forth.

Dependent claim 17, which was recited based on Stevens alone, recites:

17. A server system as recited in claim 1, wherein the presentation layer comprises:
a presentation module to determine how the replies will appear on the client devices to users; and

a rendering module, separate from the presentation module, to determine how to render the replies on the client devices.

Stevens does not disclose the subject matter recited in this claim. The outstanding Office Action addresses this claim by stating that page 250 of Stevens discloses a presentation module. As to the rendering module, the Office Action states that "Official Notice is taken that displaying information on a screen has been well known in the art for decades." While this statement, considered in a vacuum, is certainly true, it does not have bearing on the subject matter set forth in claim 17. Claim 17 states that the presentation layer (introduced in the context of the multi-layer application of claim 1) includes a presentation module and a rendering module. There is no indication in the applied references that it would have been known in the art to partition a presentation layer (in the context recited in claim 1) into the two modules recited in claim 17, where the modules have the specific roles set forth in that claim (e.g., determining how the replies will appear, and determining how to render the replies). The Office Action states that this is an inherent feature to known displays, yet none of the applied art specifically shows a presentation layer having the two distinct modules recited in claim 17.

Consider next dependent claim 22, which was also rejected under § 103 based on Stevens alone. It recites in full:

22. A server system as recited in claim 21, wherein the hierarchy of constraints comprises constraints selected from a group of constraints comprising:

- legally mandated constraints to constrain operation of the multi-layer application according to legal principles;
- company-mandated constraints to constrain operation of the multi-layer application according to preferences of a company that operates the application;
- customer constraints to constrain operation of the multi-layer application according to preferences of customers;

cultural constraints to constrain operation of the multi-layer application according to cultural aspects; and
end user constraints to constrain operation of the multi-layer application according to preferences of an end user.

Stevens nowhere discloses or even hints at a constraint hierarchy, so Stevens likewise fails to disclose the specific constraint layers recited in claim 22, comprising legally mandated constraints, company-mandated constraints, customer constraints, cultural constraints, and end user constraints. In Paragraph No. 57, the outstanding Office Action states that Stevens has “shown the use of constraints [parameters, arguments] in defining the use and limits of a program.” The Office Action further states that it “would be obvious to one of ordinary skill in the art to apply a plethora of types of constraints to the Stevens description to allow for specialized operation according to the user and system specific needs. . . .” However, the information in Steven’s configuration file does not constitute constraints that customize an application. Nor does the information in the configuration file form a hierarchy. And finally, the Examiner’s statement regarding the specific constraints recited in claim 22 has absolutely no support in the art of record, especially in view of the fact that Steven’s does not disclose *any* constraints in the context being claimed. The reasoning set forth in the rejection of claim 22 is derived from the present specification, not the prior art; this reasoning therefore reflects the use of impermissible hindsight.

Paragraph No. 30 of the outstanding Office Action again states that Stevens discloses the use of constraints in the form of parameters, and that the MPEP authorizes the Patent Office to apply a broad interpretation of the claims during examination. But claim 22 recites specific constraints, and notwithstanding the Office Action’s discussion

of parameters, Stevens does not disclose any of the specific constraints identified in claim 22.

Consider next dependent claim 25, which was rejected by the combination of Stevens and Peek. Claim 25 reads as follows:

25. A server system as recited in claim 1, wherein the presentation layer includes a form processor to generate a data input form for the multi-layer application by automatically adding, to a form definition that defines the data input form, validation code to validate subsequent inputs to one or more fields of the data input form.

Neither Stevens nor Peek disclose the above-described subject matter. The outstanding Office Action cites pages 875-879 of Peek as having a bearing on this claim. That portion describes *inter alia* a shell script for filling in forms. But this subject matter has no relationship to what is claimed other than its mention of the word “form.” For instance, neither Stevens nor Peek, whether considered alone or in combination, disclose “automatically *adding, to a form definition that defines the data input form, validation code* to validate subsequent inputs to one or more fields of the data input form.” Generally, the outstanding Office Action characterizes claims 25-28 as a “method for handling data forms.” However, closer examination of these claims reveals that they recite much more functionality than merely handling forms. In paragraph No. 32, the outstanding Office Action reasserts that Peek meets the recited features of claim 25; the Patent Office is requested to point out where Peek allegedly discloses these features.

For the above reasons, the Patent Office is respectfully requested to withdraw the rejections based on 35 U.S.C. § 103. As stated in MPEP § 2143.01, to establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). As

indicated above, all of the elements in the claims are not met by the cited references, even if, assuming *arguendo*, these references can be combined together in the manner set forth in the Office Action.

VIII. Regarding the Double Patenting Rejection

Claims 1, 17, and 48-50 were provisionally rejected under the judicially created doctrine of obviousness-type double patenting as allegedly being unpatentable over claims 1, 2, and 7-8 of copending Application No. 09/845,752 (the '752 Application). Applicant respectfully traverses this rejection for the following reasons.

Claims 1 and 48 of the present application are again reproduced for convenience of reference:

1. A server system comprising:
 - one or more computers; and
 - a multi-layer application executing on the computers to handle client requests submitted by various client devices, the multi-layer application comprising:
 - a problem-solving logic layer to process the client requests according to an associated problem domain, wherein the problem domain pertains to a particular category of service, the problem-solving logic layer containing one or more execution models to perform various sets of tasks when processing the client requests, the problem-solving logic layer producing replies to the client requests;
 - an execution environment layer to receive the client requests and select an appropriate execution model in the problem-solving logic layer for processing the client requests;
 - an interfacing layer to interface the problem-solving logic layer with one or more resources so that the execution models may utilize the resources when processing the client requests; and
 - a presentation layer to receive the replies produced by the problem-solving logic layer and to structure the replies in a manner that makes the replies presentable on the various client devices,
- wherein any of the layers may be changed without impacting other layers.

48. A method for processing client requests in a system, comprising:

receiving requests from multiple clients, the requests being related to a business-related problem domain, wherein the business-related problem domain pertains to a particular category of business-related service;

processing the requests within problem-solving logic to produce replies within the business-related problem domain, the processing comprising requesting data to be used in formulating the replies;

retrieving the data from one or more external resources and mapping the data to a domain framework for the business-related problem domain, the domain framework being independent from the problem-solving logic; and

interfacing the problem-solving logic to the domain framework to obtain the data for use in processing the request,

wherein a new business-related problem domain can be exchanged for a previous business-related problem domain by replacing one or more components of the system, without having to reconstruct an entire application solution for the new business-related problem domain.

Claims 1, 2, and 7-8 of the '752 Application, in their current form, are reproduced below:

1. A server system, comprising:
one or more computers; and
an application executing on the computers to handle client requests, the application comprising:

a business logic layer to process the client requests according to a particular business domain and produce replies to be returned to the clients in response to the client requests; and

a presentation layer separate from, but in communication with, the business logic layer to structure the replies in a manner that makes the replies presentable on different types of client devices.

2. A server system as recited in claim 1, wherein the application is reconfigurable to other business domains by substituting other business logic layers that are designed to process the client requests according to the other business domains.

7. A server system as recited in claim 1, wherein the presentation layer is configured to determine how to display the replies for a particular client.

8. A server system as recited in claim 1, wherein the presentation layer comprises:
a presentation tier to determine how the replies will appear on the client devices to users; and

a rendering tier, separate from the presentation tier, to determine how to render the replies on the client devices.

There are numerous elements in claim 1 of the present application that do not have fair counterparts in claims 1, 2, and 7-8 of the '752 Application, such as execution models, an execution environment layer, an interfacing layer, and so forth. The correlation set forth in paragraph No. 67 of the outstanding Office Action fails to establish the Office's case, as certain elements in claim 1 of the present application are not addressed, and other elements are paired up with completely dissimilar elements of the claims of the '752 Application (for instance, an interfacing layer in claim 1 of the present application cannot be construed as a request dispatcher (which is part of the presentation layer)).

Claims 48-50 of the present application are method claims, whereas claims 1, 2, 7 and 8 of the '752 Application are product-type claims. Moreover, the correlation established in paragraph No. 70 of the Office Action is erroneous. To cite merely one example, consider the element of claim 48 of the present application which recites: "retrieving the data from one or more external resources and mapping the data to a domain framework for the business-related problem domain, the domain framework being independent from the problem-solving logic." The Office Action correlates this element to a presentation layer element in the claims of the '752 Application, which is not at all relevant to the retrieving operation recited in claim 48 of the present application. Nor do the identified claims of the '752 Application mention a domain framework.

The dependent claims of the present application (17 and 49-50) are not obvious counterparts of the claims of the '752 Application at least by virtue of the deficiencies noted above with respect to independent claims 1 and 48, from which these dependent claims depend.

In conclusion, the Applicant submits that claims 1, 17, and 48-50 of the present application are not obvious counterparts of the claims of the '752 Application. For this

reason, the Applicant requests the Patent Office to remove the obviousness-type double patenting rejection of claims 1, 17, and 48-50.

The Office Action includes a number of other obviousness-type double patenting rejections. Each of these remaining double patenting rejections attempts to establish the obviousness of the claims in the present application by combining the claims of two or more copending applications. Such rejections are legally and factually misplaced. The overriding reason to make an obviousness-type double patenting rejection in a family of applications is a finding that at least two applications are basically claiming the same subject matter, such that if both applications issued as patents, the Patent Office would have granted two patents for the same concept (which is proscribed). First, the fact that a claim of the present application can be allegedly *pieced together* by identifying elements in the claims of two or more applications is not germane to whether the claims of the present application are obvious counterparts to any single other application.

It is appropriate in some circumstances to issue an obviousness-type double patenting rejection of a claim in a first application to a claim in a second application, coupled with one or more *prior art references* which establish that the differences between the two claims are obvious. *But in this case, all of the applications in question were filed on the same day, so that the claims of one application are not prior art to any of the other applications.*

For at least this reason, the Patent Office is requested to remove each of the obviousness-type double patent rejection that is structured in the legally misplaced manner described above.

IX. Regarding the Presentation of New Claim 58

This Response presents new independent claim 58, which is related to a combination of the original claims 1 and 10. This claim distinguishes over the applied art for reasons similar to those presented above with respect to claims 1 and 10.

X. Regarding the Request for Information

The Office Action makes the following request for information in the outstanding Office Action in paragraph No. 87:

Applicant is required to provide any and all available documentation concerning the development and deployment of GEtheSource (or any previous names for the aforementioned product) as mentioned in page 41 of the Applicant's remarks. Applicant is required to provide a timeline clearly stating the dates of deployment of GEtheSource (or any previous names for the aforementioned product) to provide the statutory bar for the invention.

Considering the second identified item first, undersigned is informed that the GEtheSourceProgram was beta tested and put into production after January 2001. Since this application was filed in April of 2001, the Applicant submits that these events cannot constitute Section 102-barring activity.

As to the first item, Applicant has provided one document obtained via the Internet which mentions a program referred to as GEtheSource. More generally, however, the Applicant is unsure exactly what the Patent Office is requesting and *why* it is requesting it. Considered literally, the Examiner's request encompasses any document created in the course of developing the product, including a potentially large number of

company-internal informal documents that would not be considered prior art. So that the Applicant may better comply with the needs of the Office, the Examiner is requested to explain what types of documents are needed and why they are needed.

To assist the Patent Office, the criteria used in deciding what information may be requested is set forth below, as excerpted from § 704.11 (with emphasis added):

Information which may be required under 37 CFR 1.105 is that information reasonably necessary to properly examine or treat a matter in a pending or abandoned application filed under 35 U.S.C. 111 (including a reissue application), in a pending or abandoned application that has entered the national stage under 35 U.S.C. 371, in a patent, or in a reexamination proceeding.

There must be a reasonable basis for the information required that would aid in the examination of an application or treatment of some matter. A requirement for information under 37 CFR 1.105 places a substantial burden on the applicant that is to be minimized by clearly focusing the reason for the requirement and the scope of the expected response. Thus, the scope of the requirement should be narrowly defined, and a requirement under 37 CFR 1.105 may only be made when the examiner has a reasonable basis for requiring information.

It is not presently clear what bearing documents that are not prior art would have on any rejection or objection that has been made in this application (or which could be made in this application). And therefore it is not presently clear why the Examiner's request should not be considered overly broad, unnecessarily burdensome, and therefore clearly inappropriate in view of the express instructions of the MPEP.

XI. Conclusion and Request for Examiner Interview

The arguments presented above are not exhaustive; Applicant reserves the right to present additional arguments to fortify its position. Further, Applicant reserves the right to challenge the alleged prior art status of one or more documents cited in the Office Action.

All objections and rejections raised in the Office Action having been addressed, it is respectfully submitted that the present application is in condition for allowance and such allowance is respectfully solicited.

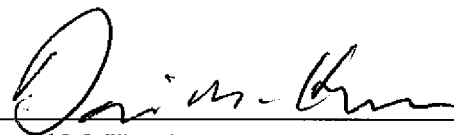
In the event that there are any issues left unresolved by this Amendment, the Examiner is requested to contact the undersigned to schedule a personal interview prior to issuance of another Office Action. The undersigned requests that an Examiner of Primary status attend the interview. The undersigned can be reached at the number listed below.

Note that the undersigned is prosecuting this case under 37 C.F.R. § 1.34(a).

Respectfully Submitted,

Dated: April 28, 2006

By: _____


David M. Huntley
Reg. No. 40,309
(509) 324-9256, ext. 248

Appendix A: Exemplary and Non-Limiting Support for Claims in Specification

Claim	Claim Element	Exemplary and Non-Limiting Support in Text	Exemplary and Non-Limiting Support in Drawings
1	A server system comprising: one or more computers; and a multi-layer application executing on the computers to handle client requests submitted by various client devices, the multi-layer application comprising:	at least page 6, line 24 to page 8, line 18	at least multi-layer architecture 110 implemented on one or more computers 112 that interact with various client devices 102
	a problem-solving logic layer to process the client requests according to an associated problem domain, wherein the problem domain pertains to a particular category of service,	at least page 10, line 25 to page 11, line 28; page 19, line 4 to page 33, line 1	at least business logic layer 204
	the problem-solving logic layer containing one or more execution models to perform various sets of tasks when processing the client requests, the problem-solving logic layer producing replies to the client requests;	at least page 11, lines 5-28; page 19, line 4 to page 33, line 1	at least one or more execution models 230; Figs. 4-6 shows one example of an execution model for an asset catalogue application
	an execution environment layer to receive the client requests and select an appropriate execution model in the problem-solving logic layer for processing the client requests;	at least page 9, line 22 to page 10, line 24	execution environment 202
	an interfacing layer to interface the problem-solving logic layer with one or more resources so that the execution models may utilize the resources when processing the client requests;	at least page 12, line 1 to page 13, line 24	note the various layers in Fig. 2 that serve at interfacing role, such as the data coordination layer 206 and the data abstraction layer 208
	and a presentation layer to receive the replies produced by the problem-solving logic layer and to structure the replies in a manner that makes the replies presentable on the various client devices,	at least page 13, line 25 to page 15, line 17; page 45, line 21 to page 50, line 16	at least presentation layer 212; Figs. 10 and 11
	wherein any of the layers may be changed without impacting other layers.	at least page 4, lines 6-10; page 6, lines 11-14; page 9, lines 11-21	note generally the layers of Fig. 2
2	A server system as recited in claim 1, wherein the execution environment layer comprises a framework to receive the client requests and route the requests to the problem-solving logic for processing.	at least page 10, lines 1-17	at least framework 220
3	A server system as recited in claim 2, wherein the execution environment layer comprises one or more adapters to interface the framework with different types of the client devices.	at least page 10, lines 18-24	at least adapters 228
4	A server system as recited in claim 1, wherein the execution environment layer comprises: a model dispatcher to route the client requests to selected execution models in the problem-solving logic layer; and	at least page 10, lines 6-17	at least model dispatcher 222
	a request dispatcher to structure the replies for return to the client devices.	at least page 10, lines 6-17	at least request dispatcher 224
5	A server system as recited in claim 1, wherein the multi-layer application can be adapted to receive requests from new client devices with incompatible communication protocols by substituting	at least page 4, lines 6-10; page 6, lines 11-14; page 9, lines 11-21	note generally the layers of Fig. 2, and particularly the role of the execution environment layer 202

Claim	Claim Element	Exemplary and Non-Limiting Support in Text	Exemplary and Non-Limiting Support in Drawings
	a new execution environment layer that supports the new client devices.		
6	A server system as recited in claim 1, wherein one of the execution models is embodied as a set of discrete program modules, each program module performing a specific task.	at least page 11, lines 5-28; page 19, line 4 to page 33, line 1	at least one or more execution models 230; Figs. 4-6 shows one example of an execution model for an asset catalogue application
7	A server system as recited in claim 1, wherein one of the execution models is embodied as an interaction-based model in which computer programs are defined by a series of interaction definitions.	at least page 11, lines 5-28; page 19, line 4 to page 33, line 1	at least one or more execution models 230; Figs. 4-6 shows one example of an execution model for an asset catalogue application
8	A server system as recited in claim 1, wherein the execution models are embodied according to at least one of a command model, an action-view model, and a use case model.	at least page 11, lines 5-28; page 19, line 4 to page 33, line 1	at least one or more execution models 230; Figs. 4-6 shows one example of an execution model for an asset catalogue application
9	A server system as recited in claim 1, wherein one of the execution models performs tasks according to a first business purpose, and the multi-layer application being reconfigurable to achieve a different business purpose by installing another execution model that performs tasks according to the second business purpose.	at least page 11, lines 5-28; page 19, line 4 to page 33, line 1; also note at least page 4, lines 6-10; page 6, lines 11-14; page 9, lines 11-21	at least one or more execution models 230; Figs. 4-6 shows one example of an execution model for an asset catalogue application
10	A server system as recited in claim 1, wherein the interfacing layer comprises: a data abstraction layer to obtain data from the resources and map the data into a domain framework that models information flow for a specific problem domain; and	at least page 12, line 1 to page 13, line 24	note at least data abstraction layer 208
	a data coordination layer that provides an interface for the problem-solving logic layer to access the domain framework of the data abstraction layer and obtain the data.	at least page 12, line 1 to page 13, line 24	note at least data coordination layer 206
11	A server system as recited in claim 10, wherein the data coordination layer comprises one or more application data managers that interface the domain framework in the data abstraction layer into an application solution space of the problem-solving logic layer.	at least page 12, line 1 to page 13, line 24	note data coordination layer 206, and particularly the role of application data managers 240
12	A server system as recited in claim 1, wherein the multi-layer application can be adapted to access new resources by substituting in a new interfacing layer that supports the new resources.	at least page 4, lines 6-10; page 6, lines 11-14; page 9, lines 11-21	note generally the layers of Fig. 2, and particularly the data coordination layer 206 and the data abstraction layer 208 which interact with the resources 108
13	A server system as recited in claim 1, wherein the client devices support different data formats, the presentation layer being configured to select appropriate data formats for encoding the replies.	at least page 13, line 25 to page 15, line 17; page 45, line 21 to page 50, line 16	at least presentation layer 212; Figs. 10 and 11
14	A server system as recited in claim 1, wherein the client devices support different communication protocols, the presentation layer being configured to select appropriate communication protocols for delivering the replies to the clients.	at least page 13, line 25 to page 15, line 17; page 45, line 21 to page 50, line 16	at least presentation layer 212; Figs. 10 and 11
15	A server system as recited in claim 1, wherein the presentation layer is configured to determine how to display	at least page 13, line 25 to page 15, line 17; page 45, line 21 to page 50, line 16	at least presentation layer 212; Figs. 10 and 11

Claim	Claim Element	Exemplary and Non-Limiting Support in Text	Exemplary and Non-Limiting Support in Drawings
	the replies for a particular client.		
16	A server system as recited in claim 1, wherein the presentation layer is configured to determine at least one of (1) a layout of individual replies, (2) display attributes in which to present the replies, and (3) a presentation theme.	at least page 13, line 25 to page 15, line 17; page 45, line 21 to page 50, line 16	at least presentation layer 212; Figs. 10 and 11
17	A server system as recited in claim 1, wherein the presentation layer comprises: a presentation module to determine how the replies will appear on the client devices to users; and a rendering module, separate from the presentation module, to determine how to render the replies on the client devices.	at least page 13, line 25 to page 15, line 17; page 45, line 21 to page 50, line 16 at least page 13, line 25 to page 15, line 17; page 45, line 21 to page 50, line 16	at least presentation layer 212; Figs. 10 and 11; note particularly presentation functionality 224 at least presentation layer 212; Figs. 10 and 11; note particularly content renderer 260
18	A server system as recited in claim 1, further comprising an authentication module to authenticate the client devices or users of the client devices.		authentication module 270
19	A server system as recited in claim 1, further comprising a constraint system to constrain operation of the multi-layer application according to a hierarchy of different constraints.	at least page 50, line 17 to page 55, line 25	at least Figs. 12 and 13
20	A server system as recited in claim 1, further comprising a constraint system to constrain operation of the multi-layer application according to multiple different constraints, the constraint system comprising a hierarchy of constraint layers, with each constraint layer containing a set of one or more constraints that customize operation of the multi-layer application.	at least page 50, line 17 to page 55, line 25	at least Figs. 12 and 13
21	A server system as recited in claim 1, further comprising: a constraint hierarchy of multiple constraint layers, each constraint layer containing a set of one or more constraints that constrain operation of the multi-layer application, the constraint layers being organized within the constraint hierarchy such that a first constraint layer limits a second constraint layer but the second constraint layer does not limit the first constraint layer; and a constraint resolver to resolve the constraint layers so that operation of the multi-layer application is constrained by a set of the constraints in the constraint layers.	at least page 50, line 17 to page 55, line 25 at least page 50, line 17 to page 55, line 25	at least Figs. 12 and 13; note particular hierarchy of constraints 1202 at least Figs. 12 and 13; note particularly constraint resolver 1204

Claim	Claim Element	Exemplary and Non-Limiting Support in Text	Exemplary and Non-Limiting Support in Drawings
22	A server system as recited in claim 21, wherein the hierarchy of constraints comprises constraints selected from a group of constraints comprising: legally mandated constraints to constrain operation of the multi-layer application according to legal principles; company-mandated constraints to constrain operation of the multi-layer application according to preferences of a company that operates the application; customer constraints to constrain operation of the multi-layer application according to preferences of customers; cultural constraints to constrain operation of the multi-layer application according to cultural aspects; and end user constraints to constrain operation of the multi-layer application according to preferences of an end user.	at least page 50, line 17 to page 55, line 25	
23	A server system as recited in claim 1, further comprising a security policy enforcement module to enforce security restrictions on accessing information stored at the one or more resources.	at least page 15, lines 18-26; page 33, line 2 to page 45, line 20	at least security policy enforcement module 280; Figs. 7-9
24	A server system as recited in claim 23, wherein the security policy enforcement module is to enforce the security restrictions based on a set of low-level security rules defined using high-level permission concepts.	at least page 15, lines 18-26; page 33, line 2 to page 45, line 20	at least security policy enforcement module 280; Figs. 7-9; note particularly Fig. 8
25	A server system as recited in claim 1, wherein the presentation layer includes a form processor to generate a data input form for the multi-layer application by automatically adding, to a form definition that defines the data input form, validation code to validate subsequent inputs to one or more fields of the data input form.	at least page 56, line 1 to page 91, line 28	at least Figs. 14-19
26	A server system as recited in claim 25, wherein the form processor is to generate the data input form by identifying one or more custom tags associated with the data input form, to replace each of the one or more custom tags with another tag, and further to add to the form definition, for each of the one or more replaced tags, validation code to validate subsequent inputs to a field corresponding to the tag.	at least page 56, line 1 to page 91, line 28	at least Figs. 14-19
27	A server system as recited in claim 25, wherein the form processor is further to automatically identify one or more data input fields to be included in the form definition.	at least page 56, line 1 to page 91, line 28	at least Figs. 14-19
28	A server system as recited in claim 25, wherein the form processor is further to automatically identify one or more restrictions associated with a data input field of the data input form, and to determine the validation code based at least in part on the one or more restrictions.	at least page 56, line 1 to page 91, line 28	at least Figs. 14-19
29	A server system as recited in claim 1, further comprising:	at least page 92, line 1 to page 101, line 15	at least Figs. 20-23

Claim	Claim Element	Exemplary and Non-Limiting Support in Text	Exemplary and Non-Limiting Support in Drawings
	a resource bundle containing locale-specific content that is authored for a particular locale;		
	and a resource bundle manager to populate a locale-independent core with the locale-sensitive content in the resource bundle to produce a computer-servable document that can be served by the multi-layer application to the particular locale.	at least page 92, line 1 to page 101, line 15	at least Figs. 20-23
30	A server system as recited in claim 29, wherein the resource bundle manager resides in the interfacing layer.	at least page 92, line 1 to page 101, line 15	at least Figs. 20-23
48	A method for processing client requests in a system, comprising: receiving requests from multiple clients, the requests being related to a business-related problem domain, wherein the business-related problem domain pertains to a particular category of business-related service;	at least page 16, lines 2-24	at least operation 302; clients 102; business logic layer 204
	processing the requests within problem-solving logic to produce replies within the business-related problem domain, the processing comprising requesting data to be used in formulating the replies; retrieving the data from one or more external resources and mapping the data to a domain framework for the business-related problem domain, the domain framework being independent from the problem-solving logic; and interfacing the problem-solving logic to the domain framework to obtain the data for use in processing the request,	at least page 17, line 23 to page 18, line 21; also note page at least page 12, line 1 to page 13, line 24	at least operations 308-314; resources 108; domain framework 250
	wherein a new business-related problem domain can be exchanged for a previous business-related problem domain by replacing one or more components of the system, without having to reconstruct an entire application solution for the new business-related problem domain.	at least page 4, lines 6-10; page 6, lines 11-14; page 9, lines 11-21	at least note generally the layers of Fig. 2
49	A method as recited in claim 48, further comprising structuring the replies for presentation to the clients.	at least page 18, line 16 to page 19, line 2; page 13, line 25 to page 15, line 17; ;page 45, line 21 to page 50, line 16	at least operations 316-320; presentation layer 212; Figs. 10 and 11
50	A method as recited in claim 48, further comprising: structuring the replies to define how the replies will appear when presented at the clients; and independent of said structuring, conforming the replies to output capabilities of the clients.	at least page 18, line 16 to page 19, line 2; page 13, line 25 to page 15, line 17; ;page 45, line 21 to page 50, line 16	at least operations 316-320; presentation layer 212; Figs. 10 and 11
51	A method as recited in claim 48, further comprising constraining how the replies are presented according to a hierarchy of constraints, wherein the hierarchy of constraints comprises multiple constraints such that a first constraint limits a second constraint but the second constraint does not limit the first constraint.	at least page 50, line 17 to page 55, line 25	at least Figs. 12 and 13
58	58. (New) A server system comprising: one or more computers; and	at least page 6, line 24 to page 8, line 18	at least multi-layer architecture 110 implemented on one or more

Claim	Claim Element	Exemplary and Non-Limiting Support in Text	Exemplary and Non-Limiting Support in Drawings
	a multi-layer application executing on the computers to handle client requests submitted by various client devices, the multi-layer application comprising:		computers 112 that interact with various client devices 102
	a problem-solving logic layer to process the client requests according to an associated problem domain, wherein the problem domain pertains to a particular category of service, the problem-solving logic layer containing one or more execution models to perform various sets of tasks when processing the client requests, the problem-solving logic layer producing replies to the client requests;	at least page 10, line 25 to page 11, line 28; page 19, line 4 to page 33, line 1	at least business logic layer 204; one or more execution models 230; Figs. 4-6 shows one example of an execution model for an asset catalogue application
	an execution environment layer to receive the client requests and select an appropriate execution model in the problem-solving logic layer for processing the client requests;	at least page 9, line 22 to page 10, line 24	execution environment 202
	an interfacing layer to interface the problem-solving logic layer with one or more resources so that the execution models may utilize the resources when processing the client requests, wherein the interfacing layer comprises: a data abstraction layer to obtain data from the resources and map the data into a domain framework that models information flow for a specific problem domain; and a data coordination layer that provides an interface for the problem-solving logic layer to access the domain framework of the data abstraction layer and obtain the data; and	at least page 12, line 1 to page 13, line 24	note the various layers in Fig. 2 that serve at interfacing role, including the data coordination layer 206 and the data abstraction layer 208
	a presentation layer to receive the replies produced by the problem-solving logic layer and to structure the replies in a manner that makes the replies presentable on the various client devices	at least page 13, line 25 to page 15, line 17; page 45, line 21 to page 50, line 16	at least presentation layer 212; Figs. 10 and 11

Appendix B: Exhibit

From: http://whatis.techtarget.com/definition/0,289893,sid9_gci343052,00.html

layer

1) In computer programming, layering is the organization of programming into separate functional components that interact in some sequential and hierarchical way, with each layer usually having an interface only to the layer above it and the layer below it. Communication programs are often layered. The reference model for communication programs, Open System Interconnection (OSI), is a layered set of protocols in which programming at both ends of a communications exchange uses an identical set of layers. In the OSI model, there are seven layers, each reflecting a different function that has to be performed in order for program-to-program communication to take place between computers.

TCP/IP is an example of a two-layer (TCP and IP) set of programs that provide transport and network address functions for Internet communication. A set of TCP/IP and other layered programs is sometimes referred to as a protocol stack.

2) In Photoshop and many other graphic applications, a layer is a component in a complex image that consists of multiple layers. Imagine a set of transparencies stacked on top of each other. Now imagine that each transparency contains part of a single image. One transparency might have the background. One transparency might have text. Another transparency might display the company logo. You can view each transparency by itself, or you can stack the transparencies on top of one another and view the stack as one image by projecting the stack on the overhead projector. It is the same with layers in a graphics application. You can work with or view each layer by itself, or you can combine them (it's called flattening) and view the "stack" of layers as one image. Layers are useful because they allow you to move and manipulate parts of an image to see how your changes affect the whole.